

Introdução

Este é um relatório sobre o trabalho realizado durante o quarto bimestre da matéria de Linguagens Formais e Autômatos (LFA) do curso de Ciência da Computação – UNESPAR (2024). Cujo o objetivo final era o de se desenvolver um software com propósito educacional para o auxílio a pessoas com dificuldade na matéria anteriormente citada.

Resumo

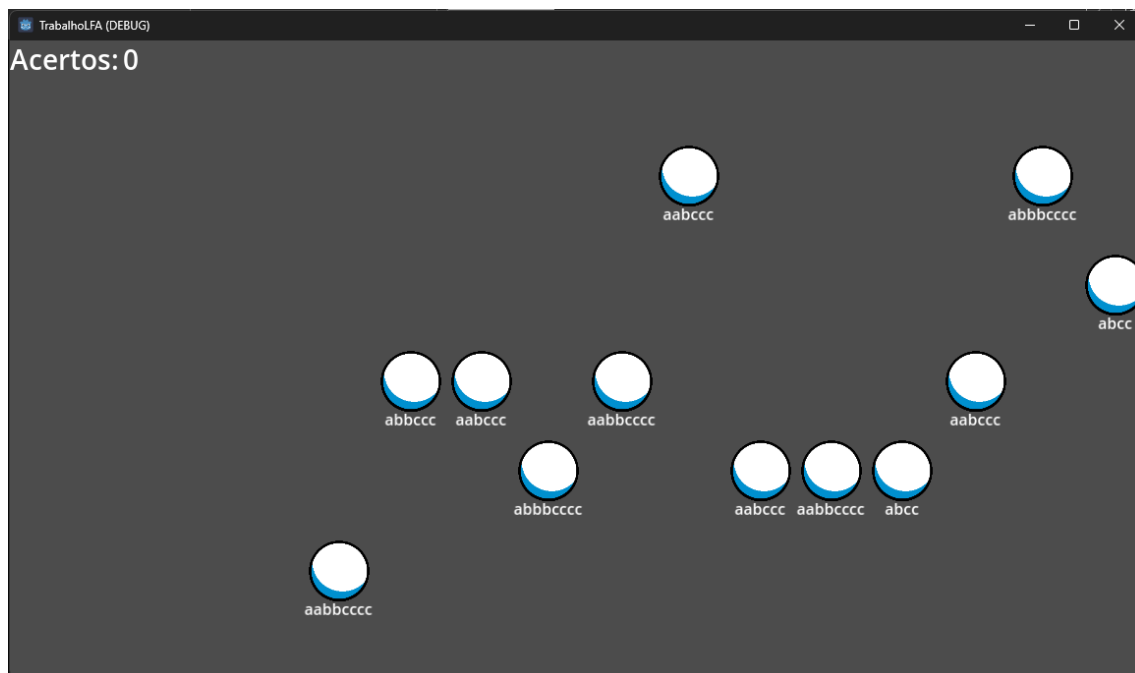
O software programado consiste em um jogo onde irão aparecer bolas que possuem palavras de linguagens regulares, o jogador deve digitar elas de acordo que aparecem, e ele deve tentar chegar o mais longe possível. À medida que a pontuação vai aumentando, a velocidade do jogo aumenta progressivamente para torná-lo mais difícil.

Caso o Jogador deixe uma das bolas alcançar o extremo lado esquerdo da tela, o jogo se encerrara, será disponibilizado a sua pontuação de acertos e uma pergunta, “Você consegue acertar qual linguagem era?”, junto de uma lista com três alternativas de linguagens distintas, sendo uma delas a linguagem usada para a partida que o jogador acabou de participar. O jogador então deve escolher alguma delas, caso escolha a errada o jogo dirá que não está certa, porém que ele deveria tentar jogar novamente para no futuro poder acertar, mas caso escolha certo, o jogo irá o parabenizar.

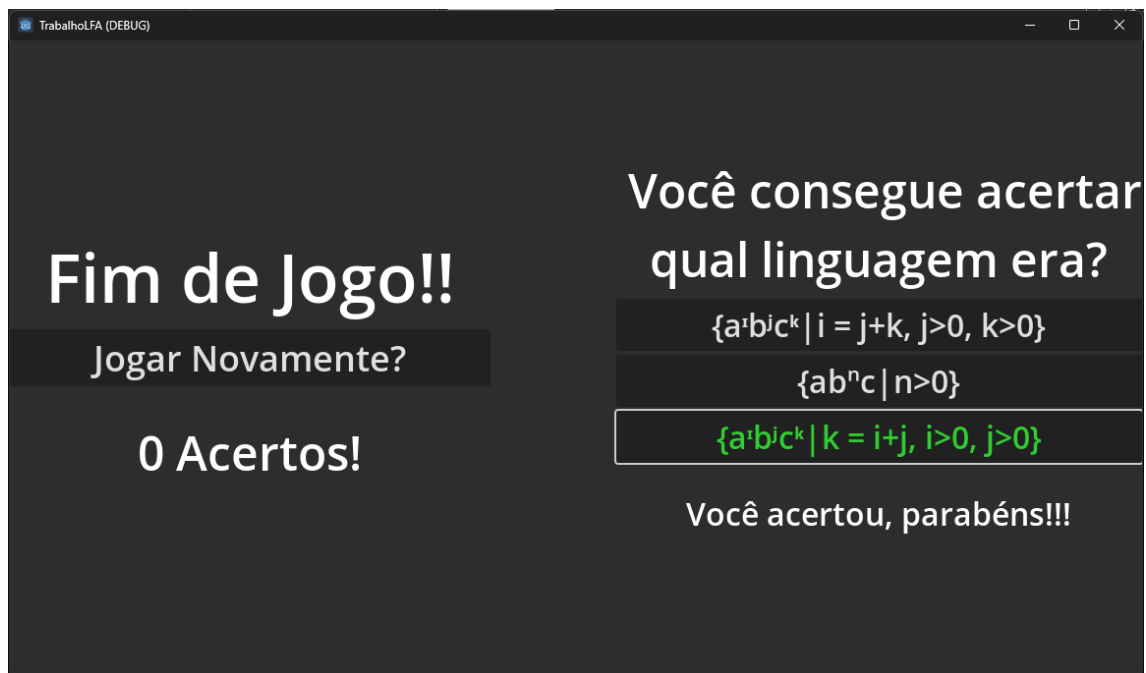
Tela Inicial:



Tela do jogo rodando, as palavras aparecem logo abaixo das bolas:



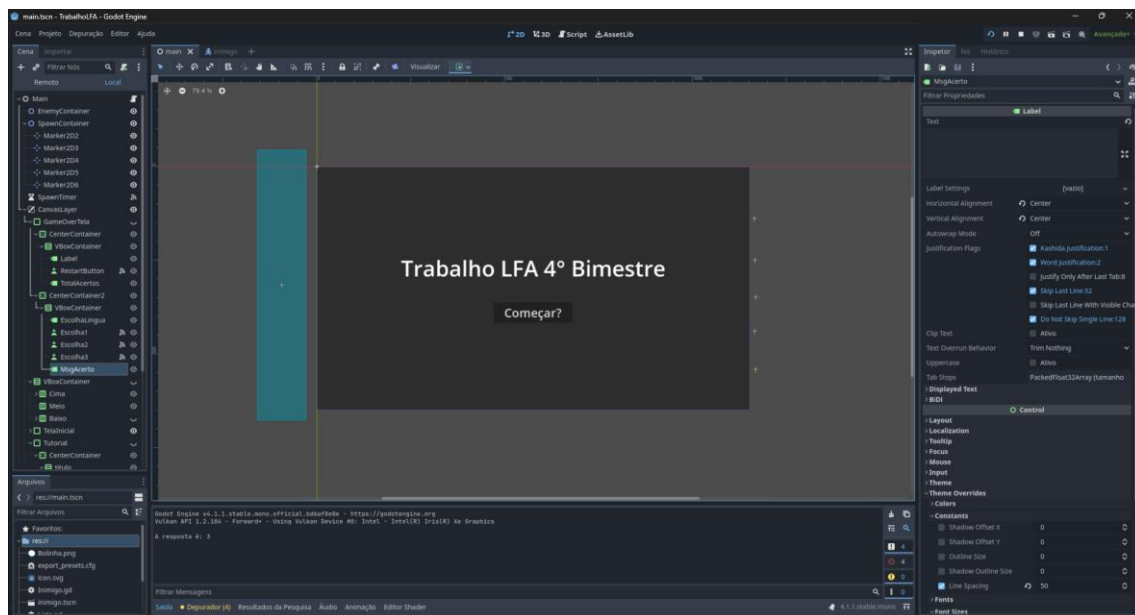
Tela de finalização do jogo com a pergunta final de qual linguagem estava sendo usada:



Materiais e Métodos

A principal ferramenta utilizada para a realização desse projeto foi a Game Engine Godot, uma ferramenta para a criação de software focado no desenvolvimento de jogos e sendo uma ferramenta OpenSource, podendo alterá-la da forma que seja de seu agrado.

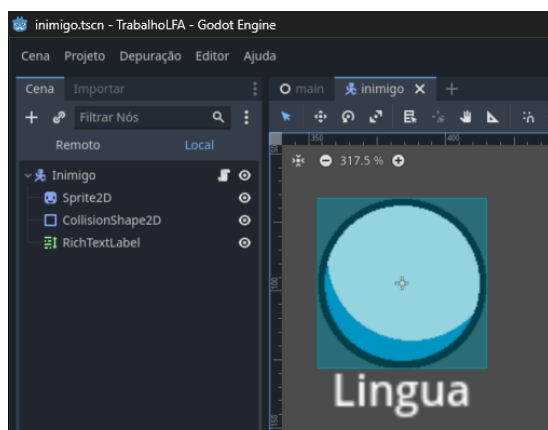
Ambiente de trabalho do programa:



A Godot possui um método interessante para guiar seus projetos de jogos tanto quanto de outros tipos de softwares interativos. Ela foca no uso do que chamamos de “Árvore” em computação. Se prestar atenção ao lado esquerdo da imagem anterior verá vários itens amontoados em cima de outros. Cada item destes é um “Node” que providencia as funções dentro da game engine.

Os Nodes na cor verde, por exemplo, são considerados nós para a manipulação de Interface Gráfica, todos os menus utilizados no projeto usaram esse tipo de nó.

Existe também o conceito de ‘Cenas’, as cenas são o que chamaríamos na programação de objetos, tendo por trás dessas cenas várias interações de nós. Por exemplo, podemos criar uma sequência de nós que nos possibilita criar um personagem, salvar a cena desse personagem e reutilizar em outra cena, cena essa que seria nosso cenário principal. É exatamente esse o conceito que é utilizado no jogo, onde temos a cena principal com as interfaces e a programação base do jogo, e temos outra cena que seria a que possui o inimigo bola.



Parâmetros utilizados e o critério para escolha do tema

Foi utilizado como parâmetro, para a realização deste projeto, o conceito de um jogo educacional, como tema de um jogo do tipo “Typing”, que seria um jogo feito para se escrever o que aparece na tela.

Um dos principais critérios estabelecido para a escolha desse tema foi o da facilidade de se implementar um jogo baseado em Linguagens Formais. Entretanto não foi a primeira ideia de projeto sugerida, pois foi feito um “Brain Storming” até chegar na decisão final.

Muitas ideias de jogos apareceram, começando com uma ideia bem desconexa com a realizada. Um jogo de cartas multijogador onde o objetivo era os jogadores desafiarem entre si utilizando o efeito dessas cartas, cada um teria um Autômato Finito (AFD), sendo o critério de vitória o primeiro a formar uma sequência de cartas em sua mão que seria computada com sucesso pelo autômato dado. Mas esta ideia foi logo descartada por sua alta dificuldade de implementação, e como eram os primeiros contatos com a ferramenta Godot, determinou-se um parâmetro de que o jogo deveria ter um escopo não tão exagerado para a sua produção no tempo determinado pelo professor.

Logo após a primeira ideia, chegou-se em uma outra que mais se assemelhava ao produto. Um jogo que apareceriam inimigos com palavras de uma certa língua e seu objetivo seria digitar essas palavras para matar esses inimigos e conquistar moedas, moedas essas que seriam utilizadas dentro de uma loja virtual onde você poderia comprar Estados de um AFD, tendo como objetivo final a montagem de um AFD capaz de processar aquela língua proposta pelos inimigos.

Conclusão

Como conclusão posso falar que este foi um trabalho que me ajudou muito a me desenvolver e conhecer a área de desenvolvimento de jogos, além também de como realmente funciona a criação de um programa.

Muitas vezes tive que voltar, ver e rever a documentação do programa, pois por ser uma ferramenta tão bem desenvolvida e com constantes mudança, maioria de seus métodos mudavam de nome, ou de conceito. Tirar dúvidas na internet, como por exemplo em vídeos, foi essencial para realização deste projeto.

Tive que aprender uma nova linguagem de programação a qual não estava acostumado, linguagem essa que é utilizada dentro do programa como sua língua principal, a GDScript. Baseada em GO e com partes semelhantes a Python, é uma excelente língua para o proposito do programa, muitas ferramentas já implementadas na língua que não existe em outras foram de grande ajuda para a parte lógica do projeto.

Exemplo da linguagem:

```
▼ func get_linguagem_array():
  » var array = []
  » array.resize(3)
  ▼ » for i in 3:                                #godot permite a interação inversa, então
  » » array[i] = linguagens[index_lin - i] #se a variavel i é negativa ela pega o size
  » return array                                #e diminui por esse i negativo dando o novo valor
```

```
54 ▼ func _unhandled_input(event):
55 ▼ » if event is InputEventKey and not event.is_pressed():
56 » » » var typed_event = event as InputEventKey
57 » » » var key_typed = typed_event.as_text_keycode()
58 » » »
59 ▼ » » if active_enemy == null:
60 » » » » find_new_active_enemy(key_typed.to_lower())
61 ▼ » » else:
62 » » » » var letra = key_typed.to_lower()
63 » » » » var prompt = active_enemy.get_prompt()
64 » » » » var next_character = prompt.substr(current_letter_index, 1)
65 ▼ » » » if letra == next_character:
66 » » » » » » ##print("success: ", letra)
67 » » » » » » current_letter_index += 1;
68 » » » » » » active_enemy.set_next_character(current_letter_index)
69 ▼ » » » » if current_letter_index == prompt.length():
70 » » » » » » » current_letter_index = -1
71 » » » » » » » active_enemy.queue_free()
72 » » » » » » » active_enemy = null
73 » » » » » » » acertos_realizados += 1
74 » » » » » » » acertos.text = str(acertos_realizados)
```

```
extends Node

var Inimigo = preload("res://inimigo.tscn")

@onready var enemy_container = $EnemyContainer
@onready var spawn_container = $SpawnContainer
@onready var spawn_timer = $SpawnTimer

@onready var lingua = $CanvasLayer/VBoxContainer/Baixo/Lingua
@onready var acertos = $CanvasLayer/VBoxContainer/Cima/ValorAcertos
@onready var fim_jogo = $CanvasLayer/GameOverTela
@onready var escore = $CanvasLayer/VBoxContainer
@onready var total_acertos_label = $CanvasLayer/GameOverTela/CenterContainer/VBoxContainer/TotalAcertos
@onready var tela_inicial = $CanvasLayer/TelaInicial
@onready var mensagem_acerto = $CanvasLayer/GameOverTela/CenterContainer2/VBoxContainer/MsgAcerto
@onready var tutorial_1 = $CanvasLayer/Tutorial
@onready var tutorial_2 = $CanvasLayer/Tutorial2

@onready var botao_1 = $CanvasLayer/GameOverTela/CenterContainer2/VBoxContainer/Escolha1
@onready var botao_2 = $CanvasLayer/GameOverTela/CenterContainer2/VBoxContainer/Escolha2
@onready var botao_3 = $CanvasLayer/GameOverTela/CenterContainer2/VBoxContainer/Escolha3

var active_enemy = null
var current_letter_index: int = -1
var acertos_realizados: int = 0

var speed_atual: float = 1.0

▼ func _ready():
  » fim_jogo.hide()
  » tela_inicial.show()
  » escore.hide()
  » #start_game()
```

```

func _ready():
    prompt_text = Lista.get_prompt()
    prompt.parse_bbcode(set_center_tags(prompt_text))

func _physics_process(delta):
    global_position.x -= speed

func get_prompt() -> String:
    return Formatar(prompt_text)

func set_next_character(next_character_index: int):
    var texto = formatar(prompt_text)
    var blue_text = get_bbcode_color_tag(blue) + texto.substr(0, next_character_index) + get_bbcode_end_color_tag()
    var green_text = get_bbcode_color_tag(green) + texto.substr(next_character_index, 1) + get_bbcode_end_color_tag()
    var red_text = ""

    if next_character_index != texto.length():
        red_text = get_bbcode_color_tag(red) + texto.substr(next_character_index + 1, texto.length() - next_character_index + 1) + get_bbcode_end_color_tag()

    prompt.parse_bbcode("[center]" + blue_text + green_text + red_text + "[/center]")

func set_center_tags(string_to_center: String):
    return "[center]" + string_to_center + "[/center]"

func get_bbcode_color_tag(color: Color) -> String:
    var cor = color.to_html(false)
    return "[color=#" + cor + "]"

func get_bbcode_end_color_tag() -> String:
    return "[/color]"

func teste() -> String:
    return get_bbcode_color_tag(blue)

func formatar(prompt_: String) -> String:
    var regex = RegEx.new()
    regex.compile("\\[.\\*?\\]")
    var text_without_tags = regex.sub(prompt_, "", true)
    return text_without_tags

```

Enfim, este projeto foi um desafio que proporcionou muitas novas habilidades e conhecimentos, sendo um projeto que já tinha a vontade de começar, mas não sabia como dar o primeiro passo. Esta área cresce cada vez mais no Brasil, e penso que é uma área necessária ter-se pelo menos um pouco de conhecimento.